Stack Algorithms at High Loads : Analysis of Unfairness or Singular Behaviours

Cédric Adjih, Philippe Jacquet, Paul Mühlethaler

▶ **To cite this version:**

Cédric Adjih, Philippe Jacquet, Paul Mühlethaler. Stack Algorithms at High Loads : Analysis of Unfairness or Singular Behaviours. [Research Report] RR-4130, INRIA. 2001. inria-00072498

HAL Id: inria-00072498
https://hal.inria.fr/inria-00072498

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Stack Algorithms at High Loads: Analysis of Unfairness or Singular Behaviours

Cédric Adjih , Philippe Jacquet , Paul Mühlethaler

No 4130

Mars 2001

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--4130--FR+ENG

Stack Algorithms at High Loads: Analysis of Unfairness or Singular Behaviours

Cédric Adjih , Philippe Jacquet , Paul Mühlethaler

Thème 1 — Réseaux et systèmes
Projet HIPERCOM

Rapport de recherche n ° 4130 — Mars 2001 — 12 pages

Abstract: In this paper, we investigate the fairness of stack access algorithms in overload conditions. We show that conventional Last In First Out free-access stack algorithms provide unfair channel access at high loads. A subset of end-users may starve during network overloads as the other nodes capture the whole channel capacity. We show that the First In First Out free-access stack protocol can cope with this unfairness problem. The blocked access stack algorithms are not showing unfairness problems.

Key-words: Stack algorithms, free access, blocked access, fairness, framing, Cable TV network (CATV network), upstream channel

(Résumé : tsvp)

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Algorithmes en Arbre à Forte Charges: Analyse de l'Équité et Comportements Singuliers

Résumé : Ce papier nous étudions l'équité des protocoles d'accès en arbre dans des conditions de surcharge. Nous montrons que l'implémentation classique (dernier arrivé premier servi) de l'algorithme d'accès en arbre à arrivée libre procure un accès inéquitable. L'accès peut être complètement interdit à des utilisateurs au bénéfice d'un sous-ensemble de ceux-ci pendant les périodes de surcharge du réseau. Nous montrons que l'implémentation de l'algorithme en arbre à arrivée libre de type premier arrivé premier servi permet de résoudre ce problème d'équité. Les protocoles en arbre à accès bloqué ne montrent pas de problème d'équité.

Mots-clé : Protocoles d'accès en arbre, accès libre, accès bloqué, équité, "framing", réseau câblé de télévision (CATV network), canal montant.

1 Introduction

A multiple access network like CATV with a large potential of simultaneously active end-users has the potentiality to get into temporary overload conditions with instantaneous demand several times larger than the channel capacity. When the total offered load is below the channel capacity, it is easy to guarantee a fair access to every end-user because there is no way to find a subset of end-users with enough ressource demand to use the whole bandwidth and denying thus any other access to other nodes.

In overload situation it is crucial to check that every active end-user obtains fair access to the available bandwidth, otherwise some end-users will load their data much faster than other, or in worst case some other end-users will never get any access. For example when the total demand is twice the channel capacity, any half of the end-users can steal and occupy the whole bandwidth, and the other half could be stay in starvation. This is a situation an appropriate MAC should avoid. Although this article concerns more praticularily the stack algorithm in the context of CATV (Cable TV) network the results are general for stack algorithms. However for a good comprehension of the results of this paper the following references can be helpful [4, 5, 6]

This paper has been proposed as a contribution to the IEEE 802.14 commitee.

2 Stack management

The stack algorithm consists into managing a virtual stack of transmitter groups. These groups are queued in the virtual stack, ranked from bottom to top. When a feedback of collision occurs, the group which caused the collision is split into three subgroups which are stored again in the queue. When a minislot is granted, a group is called for transmission and is virtually dequeued from the stack.

There are several ways to implement the stack algorithm which depend on:

1. free or blocked access;
2. stack management.

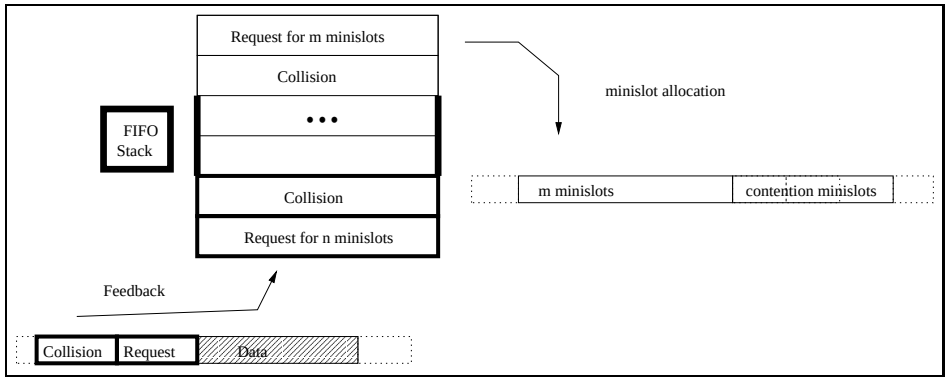
We have already analysed the stack algorithms in free access and blocked access [7]. In this paper we investigate the stack management. Although they are plenty options in stack management we restrict ourselves on the alternative:

- the Last In First Out (LIFO) management;
- the First In First Out (FIFO) management.

The LIFO management is the most conventional management for stack algorithm in the literature, [1, 2, 3, 8, 9]. It consists into granting minislot to groups from bottom, and to queueing splitted groups starting from bottom too. In particular LIFO allows distributed management without permanent monitoring, since one does not need to know *a priori* the size of the stack at feedback time in order to reschedule the retransmissions (one only need to monitor the sequence of events "collision" and "non-collision" between retransmissions). Least but not last, when propagation delay is zero, LIFO exact delay analysis is tractable.

The FIFO management is less conventional. It consists into granting minislots to groups starting from top, and to queueing splitted groups from bottom. FIFO does not allow limited sensing distributed management since one must *a priori* know the size of the stack in order to correctly schedule retransmissions.

In our implementation, the contention group FIFO stack was merged with the data request stack . The following example shows how the stack evolves with time:



Since there is a one-to-one mapping between the FIFO stack and the minislot allocation, there exists a straightforward implementation of this algorithm in single interleave:

- *C1*) The algorithm keeps a pointer (counter) to the next first non-allocated minislot;
- *C2*) All received requests and collisions are granted (contiguous) minislots at once, beginning at the "counter of the first non-allocated slot";
- *C3*) If there are unallocated minislots that would be wasted the head-end allocates them as root group contention minislots ($RQ = 0$).

3 Performance

The simulations use the parameters of table 1.

Simulations parameters	Values
Number of active stations	200
Distance from stations to headend	200 km (ranging)
Downstream data transmission rate	infinite
Upstream data transmission rates	3 Mbits/s
Propagation delay	5 μs/km
Length of simulation	30 seconds
Guardband and pre-amble	duration of 5 bytes
Minislot size	16 bytes
Headend processing delay	0 ms
Headend processing scheme	feedback and allocation at each minislot

Table 1: Network Simulation Configuration Parameters

We also use the following assumptions: implicit framing, no piggy-backing, R computation, Poisson trafic. The size of the packets follows a random distribution which is given in table 2.

packet length in bits	probability
512	0.6
1024	0.06
2048	0.04
4096	0.02
8192	0.25
12144	0.03

Table 2: Packet length distribution

The results of the simulations here use the same framework and same

It is not easy to detect unfairness unless by tracing the channel utilization of every end-user ; thus the total number of bytes transmitted by each of the 200 stations was logged and is reported in the following graphics. In the graphics, the 200 values, are represented, sorted by increasing values. When the protocol is fair, few differences can be spotted between the stations, and the curve is very close to an horizontal line. When unfairness occurs, the curve is stepper. When the unfairness leads to some end-users starvation, the curve remains on zero throughput as long there are starving users and shows a sharp step when the first non-starving node is encountered.

The graphics are reported for the following amounts of traffic generation: 1.45 Mb/s (high but non overloaded conditions), 2.42 Mb/s (overloaded), and 4.84 Mb/s (overloaded), and for the following algorithms: FIFO blocked access with entry persistence R, FIFO free access with R, LIFO blocked access with R, LIFO free access with R, and LIFO free access without R.

3.1 Performance in non overloaded conditions

In high traffic but non overloaded conditions (1.45 Mb/s, mean delay roughly equal to 40 ms), all the algorithms perform comparatively with respect to fairness.

The differences of throughput between that stations observed are in part accountable to natural differences of number of generated packets, as illustrated in the following figure 2

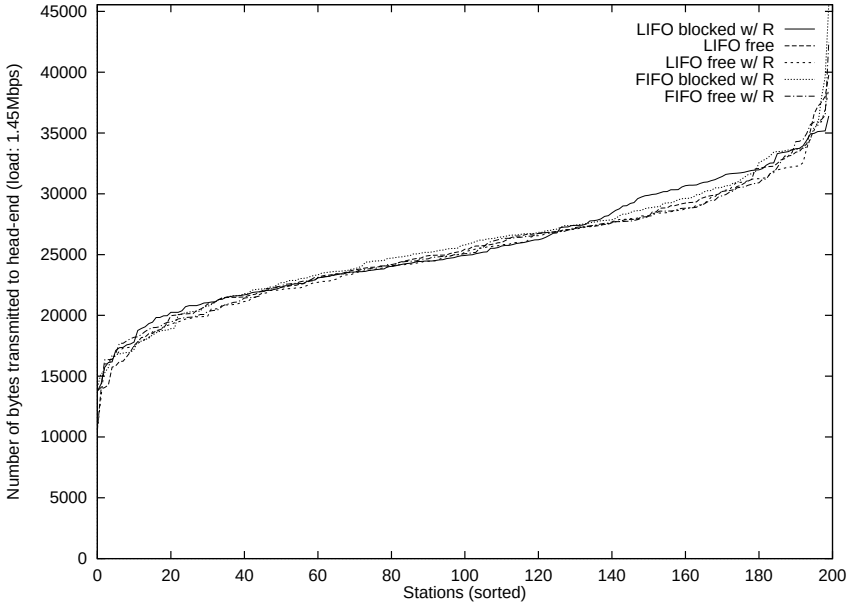


Figure 1: Number of transmitted bytes per station (stations are sorted), input load 1.4 Mbps.

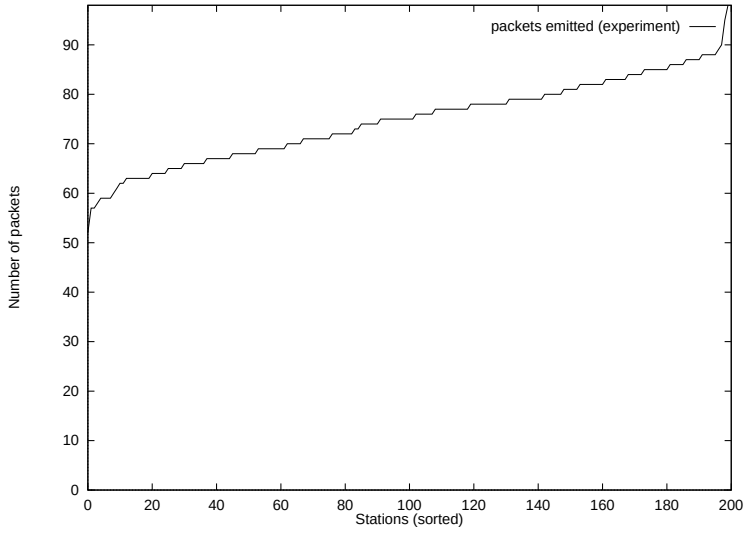


Figure 2: Number of generated packets per station.

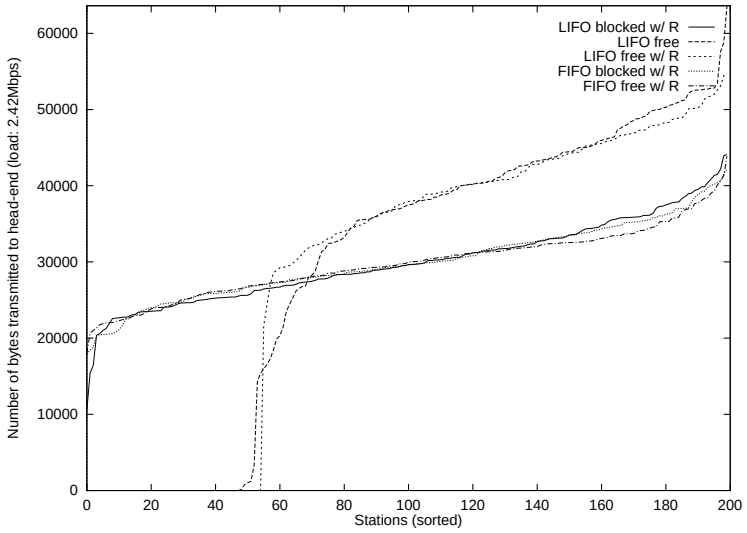


Figure 3: Number of transmitted bytes per station (stations are sorted) input load 2.4 Mbps.

It indicates the number of packets *generated* by 200 poisson sources (equivalent to 200 sources type 1), for 30 seconds, in a simple experiment.

3.2 Performance in overloaded conditions: unfairness occurence

In overloaded conditions, the FIFO algorithms and the LIFO algorithms with blocked access perform still closely, and no significant unfairness is observed.

But it appears that in the LIFO case with free access (with or without R), unfairness occurs, for a load of about 2.4 Mbps: some stations (about 45) are unable to transmit.

If the load is doubled (second graph, 4.8 Mbps), the number of stations unable to transmit is also dramatically increased (about 110).

In the following two sections, we attempt to give a quick insight of the behavior of the different free access algorithms.

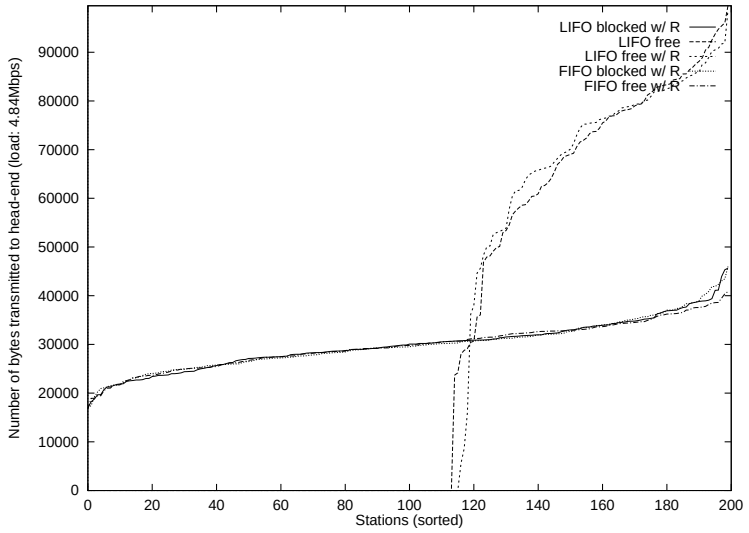


Figure 4: Number of transmitted bytes per station (stations are sorted) input load 4.8 Mbps.

3.2.1 Free access behavior

In infinite poisson population, for the two overload conditions outlined here, the free access resolution would be instable, and the stacks would grow infinitly. With finite population, intuitively, the stack can grow enough thus hold enough colliders, so that only a limited number of stations are left out, and reach a state of equilibrium.

With a free entry LIFO stack, stations at the top of the stack are infinitely delayed, and a limited number of stations monopolizes the upstream channel: in high load, each time a station receives a grant, it still has packets to transmit with high probability, and makes another request, preventing the stack to shrink. This phenomenon leads to the shown unfairness.

With free entry FIFO stack, a kind of round-robin (TDMA) emerges. On a large time scale, every station not in the stack will ultimately collide, and every station in the FIFO stack will manage to leave the stack, and thus after a random number of collisions will made a successful request. Therefore the stations would fairly leave or enter the stack.

On smaller time scale, when the stack is around its steady-state mean size, stations out of the stack might get unfair advantages: as long as they do not collide, they would be able to make successful requests. Our FIFO implementations prevent this, since the collision and the data request stacks are merged: the grant for a successful station is delayed until all previous collision groups are allocated contention slots, very similarly to low load conditions.

The drawback of FIFO implementations is the number of different RQ values, since potentially all the stations could be in collision groups in the stack. The number for bytes for RQ parameter should be large enough: the theoretical bound for RQ is proportional to the number of stations.

We did not implement packet concatenation in our simulation, but we still believe that concatenation does not prevent unfairness and may ultimately increase unfairness. With bounded concatenation the protocol is still unfair since data allocation is finite per user. With unbounded concatenation unfairness would occur but for other reason than collision resolution since a subset of users could steal the whole bandwidth with few request.

3.2.2 Blocked access behavior

For blocked access, new requests are prevented to use the upstream channel until the stack empties. This property prevents the bandwidth to be stolen by a subset of users unless unbounded concatenation is allowed. Furthermore any station in any collision group will ultimately be able to make a successful request unless it enters in indefinitely repeated collision within its group which happens with probability zero. In the examples of the simulations the blocked algorithm appears further to be as fair as free access FIFO algorithms.

4 Conclusion

In this article, we have analyzed the behavior of various implicit framing contention resolution algorithms. Simulation results show that, under high input loads, unfairness occurs with the LIFO stack free access algorithms but that this unfairness is not present with the FIFO stack free access algorithm. There is significant unfairness with the stack blocked access algorithms in LIFO or

FIFO mode. Simulation results also show that the use of a dynamic persistence parameter does not influence the fairness.

We also shown that some variants of FIFO algorithms are very simple to implement. Simplicity is likely to be an advantage when additional constraints (for example for quality of service) are to be integrated.

References

[1] J. L. Massey, "Collision Resolution Algorithms and Random-Access Communications", in *Multi-User Communication Systems*, G. Longo Editor, *CISM Courses and Lectures no.* **255**, Springer Verlag, Wien-New York, 1981, 73-137.

[2] B. S. Tsybakov, V. A. Mikhailov, "Free Synchronous Packet Access in a Broadcast Channel with Feedback", *Probl. Inform. Transmission* **14**, 1979, 259-280.

[3] P. Mathys, P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with or blocked channel access," in *IEEE Trans. on Information Theory*, vol IT-31, pp 217-243, 1985.

[4] P. Jacquet, P. Mühlethaler, P. Robert, "CATV slotted multiple access MAC," INRIA Research Report 4106, 17 p., January 2001.

[5] P. Jacquet, P. Mühlethaler, P. Robert, "Performant implementation of tree collision resolution on CATV network," INRIA Research Report 4107, 34 p, February 2001.

[6] P. Jacquet, P. Mühlethaler, P. Robert, "Framing protocols on upstream channel in CATV networks: asymptotic delay analysis," INRIA Research Report 22 p., to appear 2001.

[7] C. Adjih, P. Jacquet, P. Mühlethaler, "Stack Algorithms in implicit framing, free access, and blocked access", INRIA Research Report 19 p., 2001 to appear.

[8] G. Fayolle, P. Flajolet, M. Hofri, P. Jacquet, "Analysis of a Stack Algorithm for random multiple-access communication," *IEEE Trans. Inform. Theory*, vol IT-31, pp. 244-254, 1985.

[9] L. Merakos, C. Bisdikian, "Delay analysis of the n-ary stack random-access algorithm," in *IEEE Trans. on Information Theory*, vol IT-34, pp 931-942, 1988.

[10] P. Jacquet, P. Mühlethaler, "Minimac : a high speed access protocol based on a free access tree algorithm," INRIA RR-0849, 22 p., 1988.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399